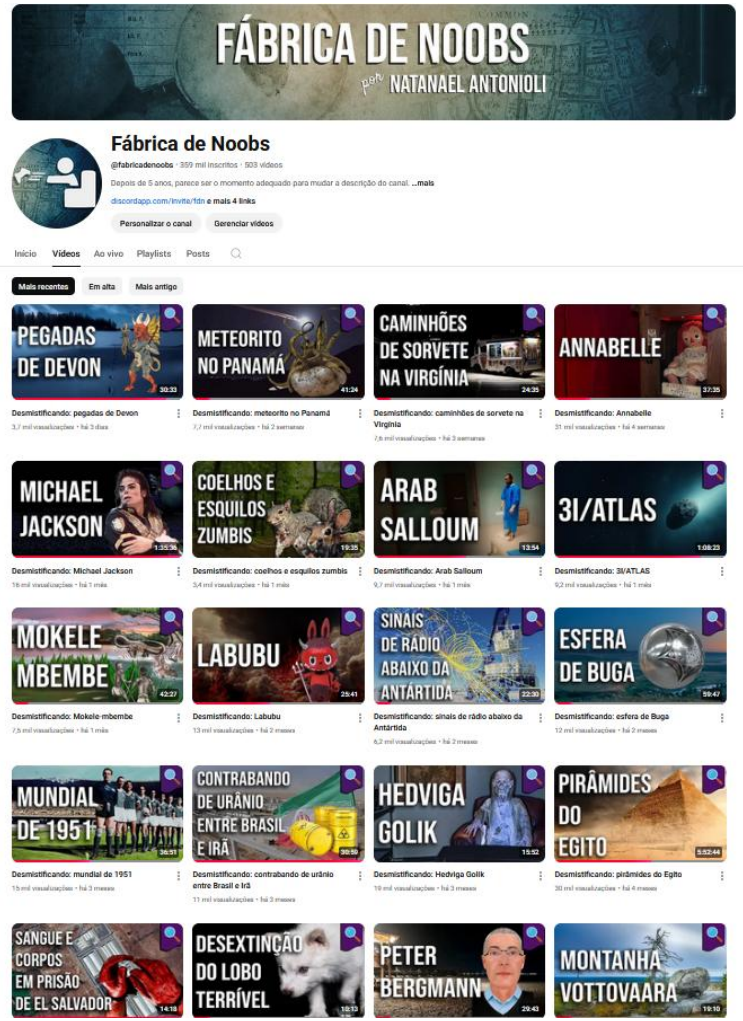
Criptografia e aplicações: do passado ao futuro

Natanael Antonioli

Outubro de 2025

Quem sou?

- Graduando em Engenharia da Computação pela Universidade Federal de São Carlos (UFSCar).
- Criador do canal **Fábrica de Noobs** no YouTube.
- Autor da série **Desmistificando**, em que mistérios do mundo real são analisados utilizando um paradigma racional e científico.
- Autor de minicursos no YouTube em áreas da computação como lógica digital, lógica de programação e criptografia.



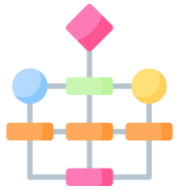
Terminologia na criptografia

Texto limpo

Texto original, que pode ser lido por qualquer pessoa que o intercepte.

Texto cifrado

Texto resultante de um método de criptografia, que só pode ser lido por quem conheça a forma de decifrá-lo.



Cifra

Algoritmo utilizado para converter um texto limpo para um texto cifrado, opcionalmente, com o uso de informações adicionais.

Chave

Informações adicionais utilizadas pela cifra.

Criptoanálise

Arte de quebrar um texto cifrado sem possuir a cifra e a chave.

Objetivos

A criptografia é empregada em múltiplos sistemas e, geralmente, seu uso visa atingir um ou mais dos seguintes objetivos.

- **Confidencialidade**: garantir que o conteúdo de uma mensagem seja ocultado ao codificá-la.
- **Integridade**: garantir que a mensagem recebida é a mesma mensagem que foi enviada, protegendo contra alterações acidentais e maliciosas.
- **Autenticação**: garantir que um usuário possa provar sua identidade para outro.

Definição

Cada porção do texto limpo é convertida, usando uma cifra, em uma porção cifrada.

São relativamente simples de se quebrar com computadores pessoais modernos:

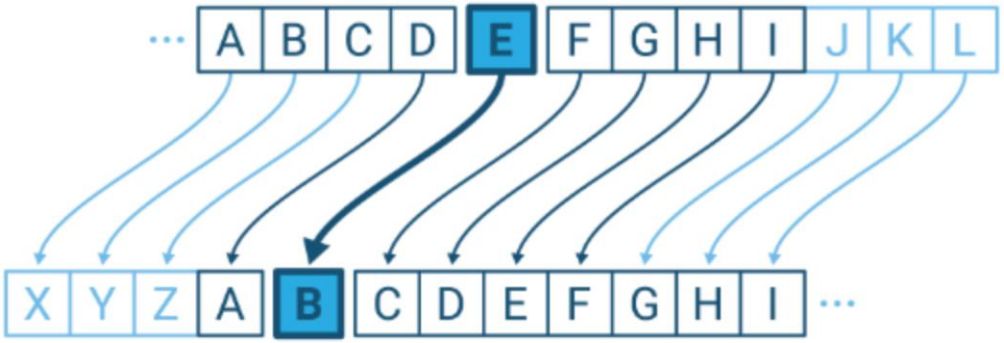
- Ataques de força bruta;
- Ataques de frequência de caracteres;
- Mera dedução através da intuição e da tentativa e erro.

Muito usadas na antiguidade em comunicações secretas, especialmente em tempos de guerra.

No presente, inviável para qualquer aplicação segura. Usada para enigmas e entretenimento.

Cifras de substituição simples

Cada letra presente na mensagem é mapeada para outra letra do alfabeto conforme um algoritmo.



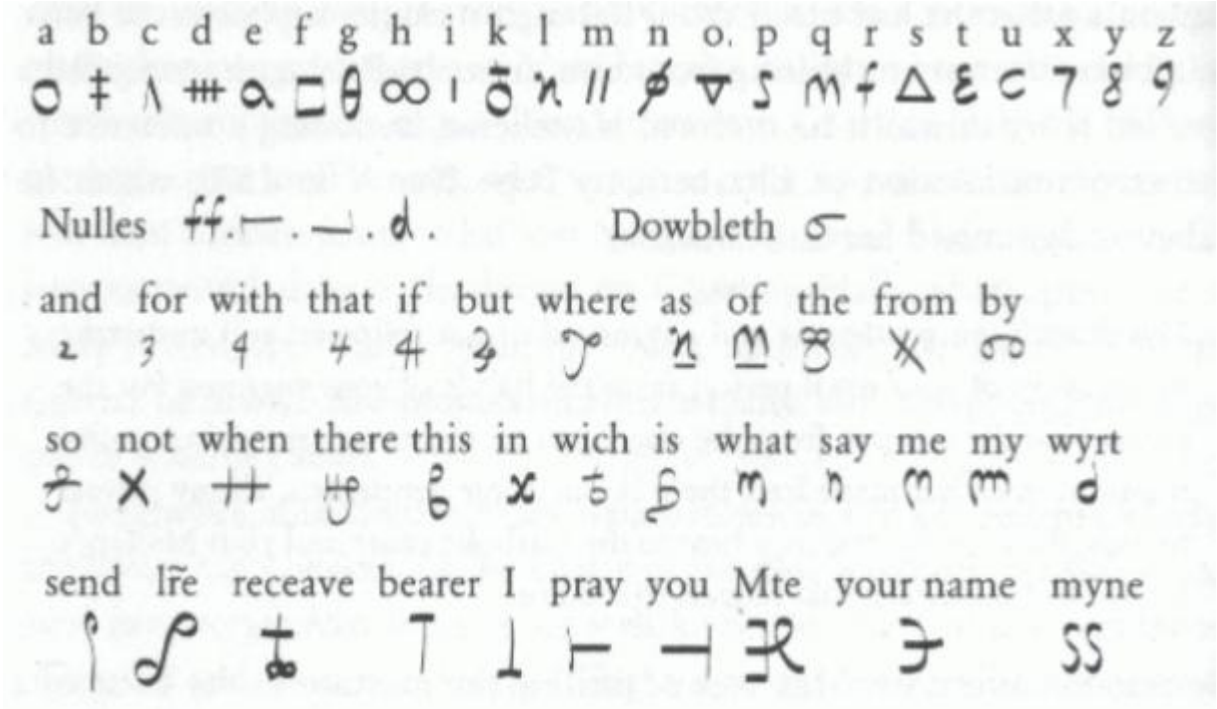
- Na **Cifra de César**, o alfabeto é deslocado em k posições, sendo k a chave da cifra.
- Em **ROT13**, tem-se uma Cifra de César com $k = 13$.
- Na **Cifra de Atbash**, o alfabeto é completamente invertido.

Cifras de substituição *nomenclator*

Nomes próprios são codificados com uma cifra simples.

Preposições, assim como verbos, substantivos e adjetivos que possuem significados importantes são codificados por símbolos específicos.

- Cada atividade **usa a própria cifra**.
- Usada entre os séculos XV e XVIII para espionagem, diplomacia e conspirações.
- Ao lado, a cifra utilizada por Maria I, da Escócia, quando tramou contra Isabel I, da Inglaterra, na disputa pelo trono inglês.


a b c d e f g h i k l m n o. p q r s t u x y z
Nulles
Dowbleth
and for with that if but where as of the from by
so not when there this in wich is what say me my wyrt
send lr̃e receave bearer I pray you Mte your name myne

Cifras de substituição homofônicas

Para tornar ataques mais difíceis, uma mesma letra mapeia para várias letras a depender de sua posição.

- Na **Cifra Literária**, a mensagem cifrada contém números. O número X mapeia para a X-ésima posição de um livro em particular.

```
5 6 13 11 22 12 17 3 2 32 21 37 34

I was destined to this vantage point
Which is so far from the Sea

I wa(3)s d(5)e(6)stine(11)d(12) f(13)or th(17)is va(21)n(22)tage
point

W(32)hi(34)ch i(37)s so far from the Sea

DEFEND HAWAII
```

Cifras de substituição polialfabéticas

Ainda temos uma cifra de substituição mas, aqui, múltiplos alfabetos são utilizados a depender da posição da porção do texto limpo. Passamos a usar uma tabela para codificação e decodificação, chamada *tabula*.

- Na **cifra de Vigenère**, a chave é uma palavra de tamanho p. A n-ésima letra da mensagem é codificada com uma cifra de César cuja chave é a $(n \bmod p)$-ésima letra da palavra.

- A **cifra de chave corrente** usa uma chave muito longa, como uma passagem de texto, para evitar a repetição da cifra de Vigenère.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Cifras mecânicas

Diversas máquinas foram usadas para implementar sistemas de criptografia, em geral, de cifras polialfabéticas.

A **Roda de Jefferson** era usada para implementar uma cifra de Vigenère com chave aleatória.

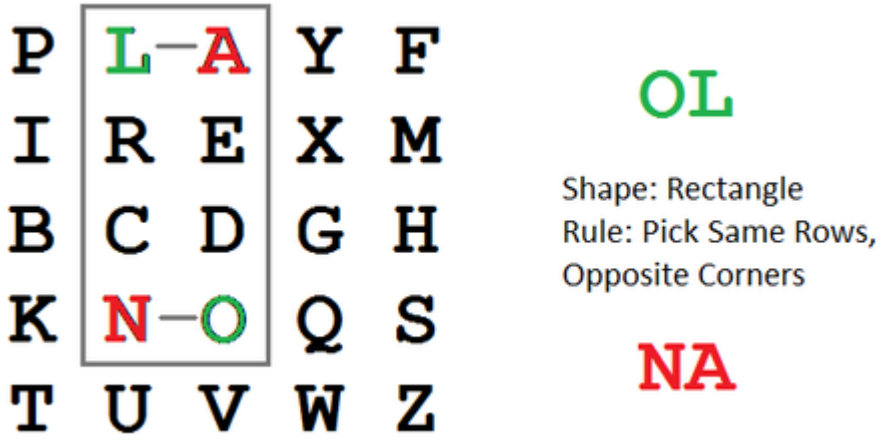
A **Máquina Enigma** era configurada pela posição de múltiplos rotores, baseada em chaves que mudavam diariamente, e implementava uma cifra de substituição usando teclas e luzes.

Cifras de substituição poligráficas

Cifras de substituição polialfabéticas são vulneráveis porque, a menos que uma chave muito grande seja usada, a repetição é facilmente detectável.

Cifras de substituição poligráficas substituem n-gramas de letras. Uma tábula para uma cifra poligráfica tem 26^n bigramas – é muito mais difícil quebrar uma cifra com 676 elementos do que uma com 26.

- Na **cifra Playfair**, que possui $n = 2$, uma regra define como escolher os pares em um quadrado 5×5, em que duas letras (como I e J) têm o mesmo valor.



Números aleatórios

Gerar uma chave aleatória é outro desafio, já que geradores de números pseudoaleatórios comuns não são seguros.

Para isso, usamos um **gerador de números pseudoaleatórios criptograficamente seguro** (CSPRNG).

- Teste do próximo bit: dados os k primeiros bits, não deve existir um algoritmo com eficiência polinomial capaz de prever o bit seguinte com taxa de sucesso maior que 50%.

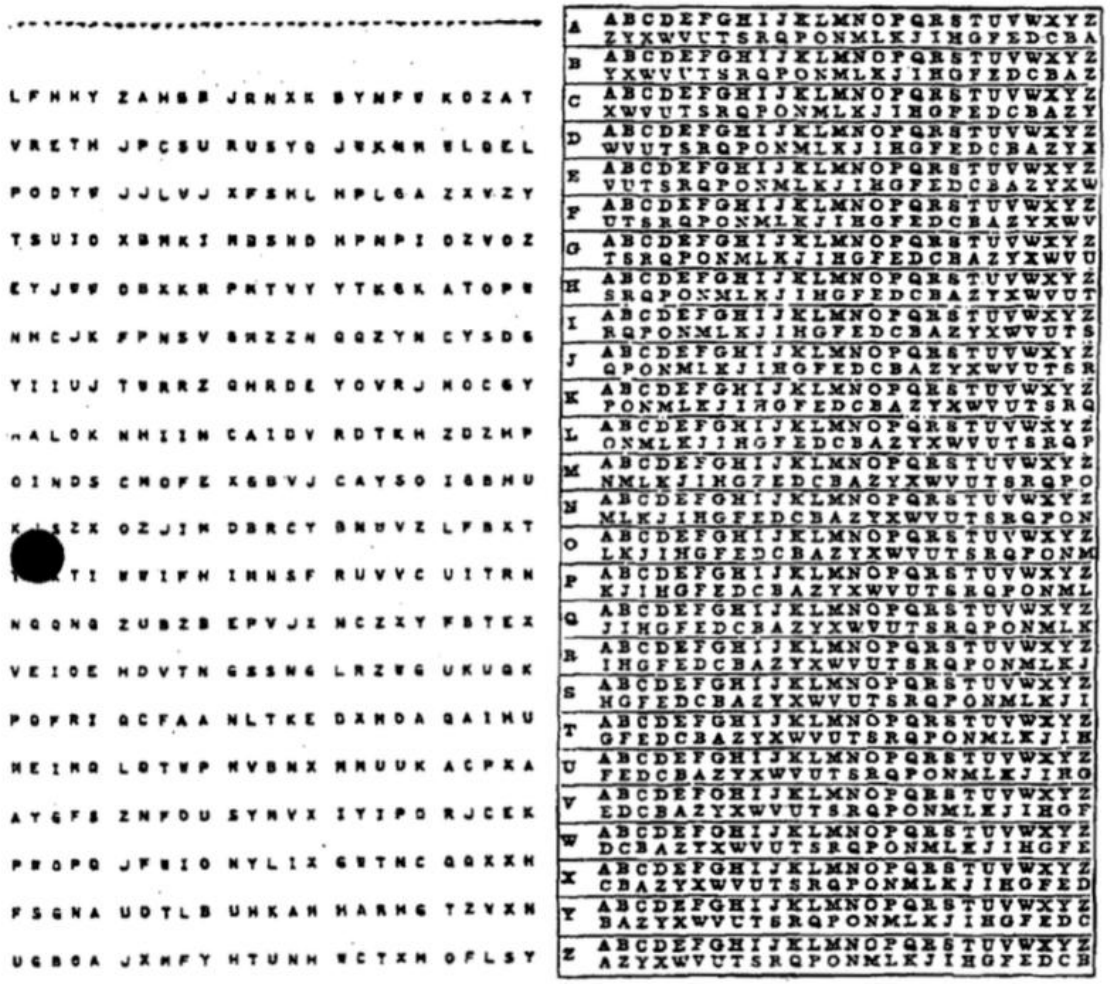
- Comprometimentos de estado: se o estado atual do gerador for revelado total ou parcialmente, deve ser impossível reconstruir os números gerados antes da revelação.

One-time pad

Um ***one-time pad*** implementa uma cifra de substituição polialfabética que é matematicamente indecifrável desde que:

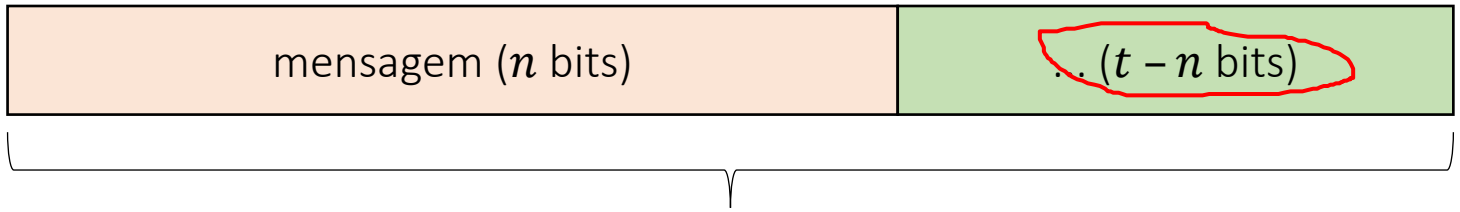
- A chave seja maior que a mensagem;
- A chave nunca seja reaproveitada em outras mensagens;
- A chave seja aleatória;
- A chave seja mantida em segredo por ambas as partes.

A última exigência é difícil de ser garantida e, por isso, chaves assimétricas são necessárias.



Blocos

Cifras de bloco recebem um texto limpo e adicionam bits até que seu tamanho seja múltiplo de um valor fixo.



$$n + (t - n) \equiv 0 \bmod t$$

Então, um algoritmo recebe os blocos de texto limpo e produz um texto cifrado de mesmo tamanho.

Cada bloco é cifrado usando a mesma chave, e algoritmos seguros realizam operações entre os blocos.

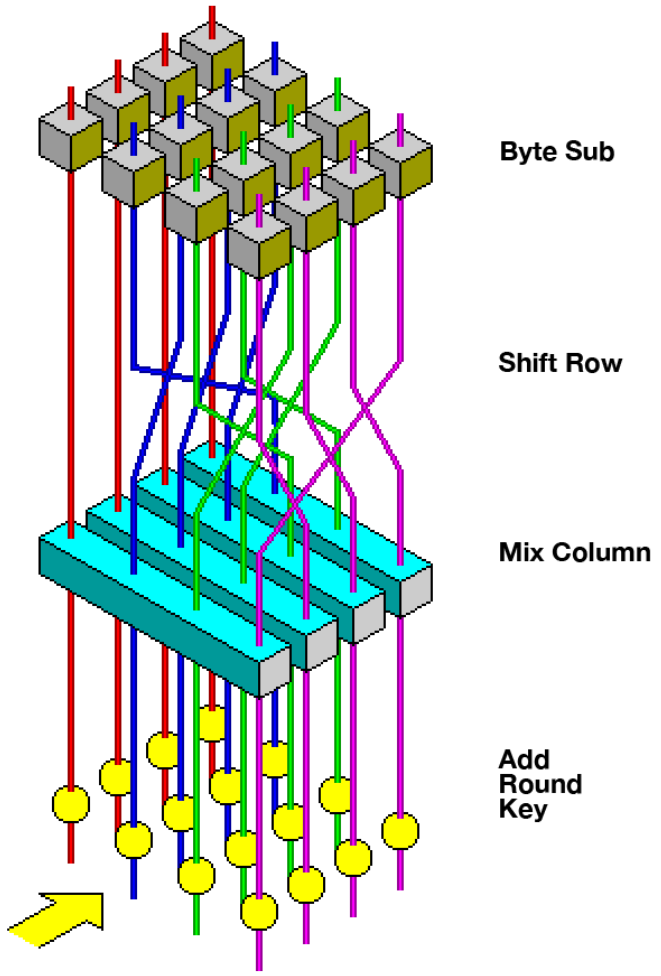
Para decifrar, um algoritmo reverte o processo com a mesma chave.

Ao contrário de cifras de substituição clássicas, cifras de bloco são seguras para certas aplicações no presente.

Advanced Encryption Standard (AES)

A mensagem é transformada em blocos de 128 bits e, em seguida, cada bloco de 16 bytes é representado em uma matriz 4x4 e passa por certo número de *rounds*, com 4 operações:

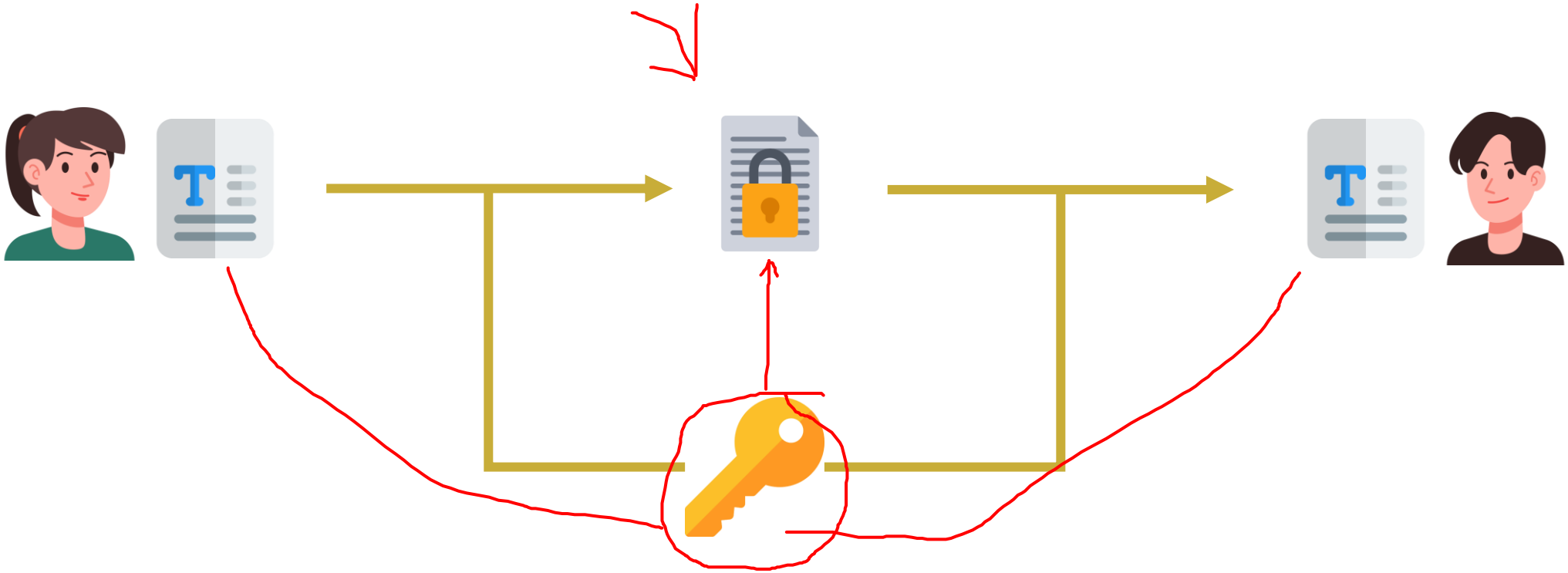
- *Byte sub*, em que bytes são substituídos conforme uma tabela de substituição dependente da chave.
- *Shift row*, em que linhas da matriz são deslocadas em certo número de posições.
- *Mix column*, em que os 4 bytes de cada coluna são misturados em operações matriciais (não realizada no último round).
- *Add round key*, em que uma chave circular derivada da chave principal é combinada com cada bloco.



Simetria

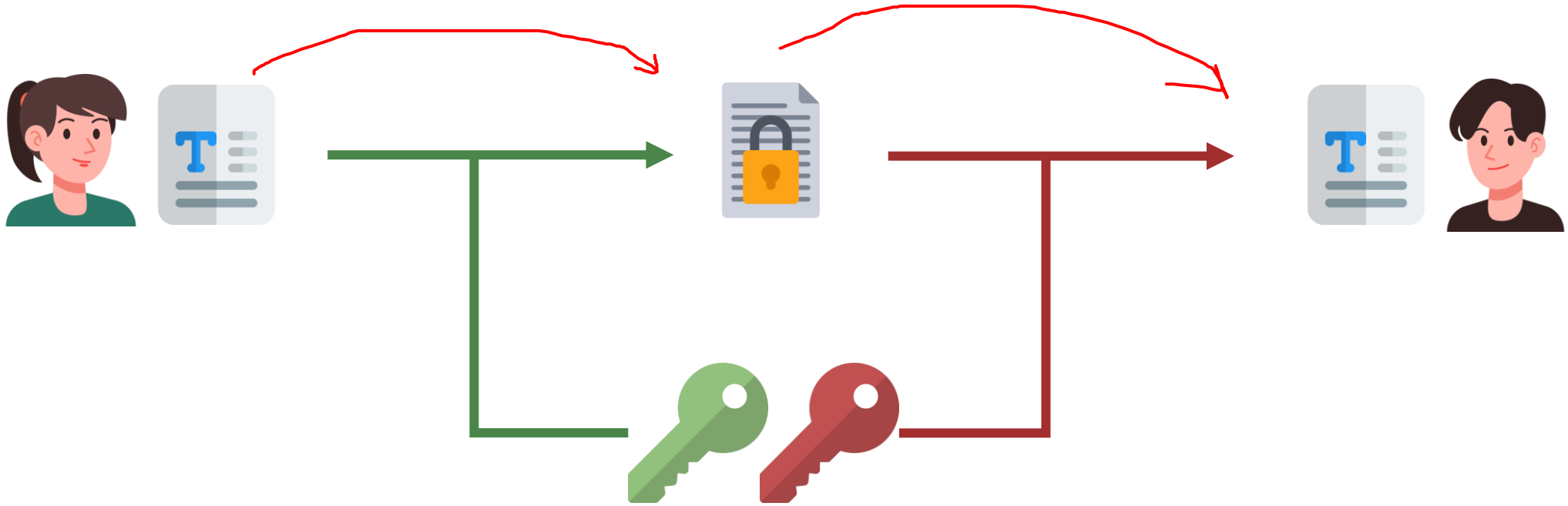
AES é segura, desde que a chave nunca seja transmitida através da rede.

E isso é um grande problema.



Assimetria

Na criptografia de chave assimétrica, a chave pública permite codificar qualquer mensagem, mas não permite decifrar nenhuma. A chave privada, por outro lado, permite decodificar qualquer mensagem.



Assim, a chave privada nunca é transmitida. A chave pública é inútil na mão de um atacante. A cifra é segura quando é computacionalmente inviável obter a chave privada a partir da chave pública.

Criptografia Rivest–Shamir–Adleman (RSA)

As chaves pública e privada precisam ser geradas algoritmicamente, e um bom algoritmo precisa ser resistente a ataques de força bruta. Isso é proposto com a **criptografia RSA**.

Pré-inicialização

1. Gere dois grandes números primos p e q.
2. Compute $n = p \times q$,
3. Compute $\varphi(n) = (p-1)(q-1)$,
4. Escolha e tal que $1 < e < \varphi(n)$ e $\text{mdc}(e, \varphi(n)) = 1$.
5. Compute d tal que $d \times e \equiv 1 \bmod \varphi(n)$,

Chaves

1. Chave privada será (d, n)
2. Chave pública será (e, n)

Codificação

1. Transforme a mensagem em um número m.
2. Encripte a mensagem com $c = m^e \bmod n$.

Decodificação

1. Decripte a mensagem com $m = c^d \bmod n$.
2. Limpe a mensagem original.

Limpar?

A mensagem pode ser decifrada usando valores ASCII e expoentes. "ABC" viraria:

$$65 \times 256^2 + 66 \times 256^1 + 67 \times 256^0$$

Mas RSA é determinístico isso é inseguro.

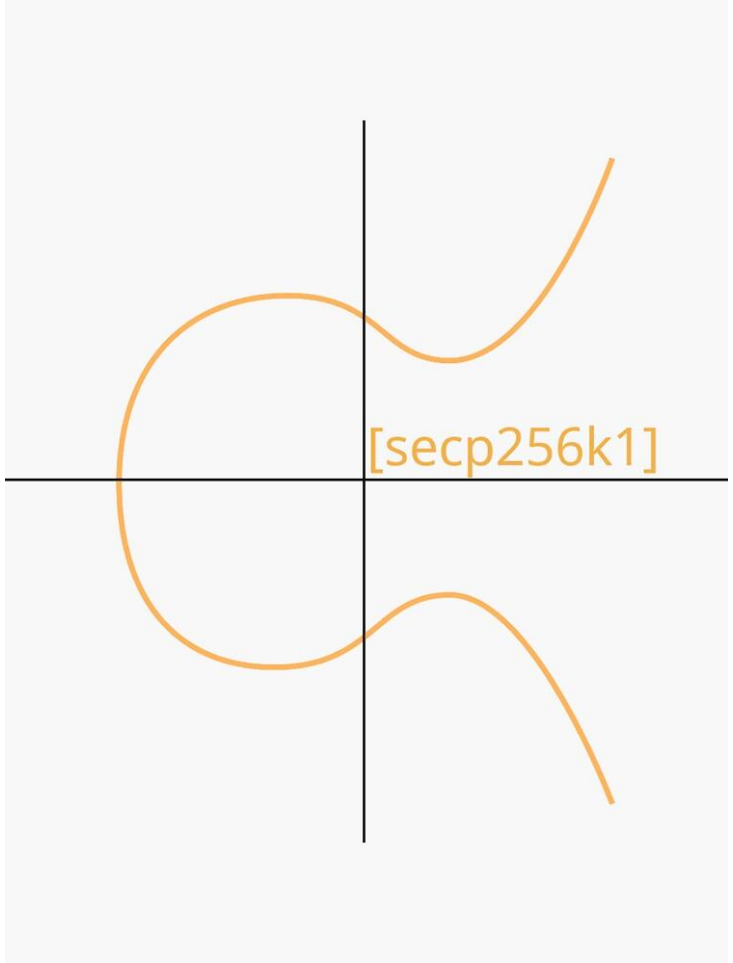
Então, adicionamos números aleatórios. Alguém que queira quebrar a mensagem precisará adivinhar esses números.

Criptografia de curva elíptica

A **criptografia de curva elíptica** ainda é estudada, mas pode fornecer chaves menores que as da criptografia RSA com a mesma segurança. Para isso, define-se:

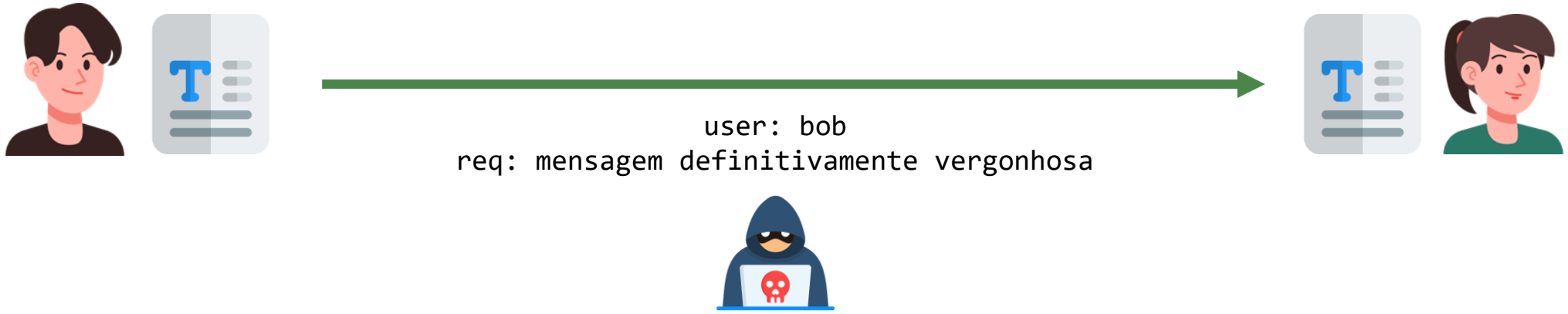
- Um campo finito $\mathbb{F}_p = 0,1,2, \ldots p$, sendo p um primo grande;
- Uma curva elíptica dentro desse plano, da forma $y^3 = x^3 + ax + b$;
- Um ponto arbitrário da curva $G(g_x, g_y)$;
- Um número aleatório k como chave privada;
- Uma chave pública dada por $P(p_x, p_y) = kG$

Se um algoritmo codifica texto em pontos na curva, então pode-se cifrar e decifrar mensagens.

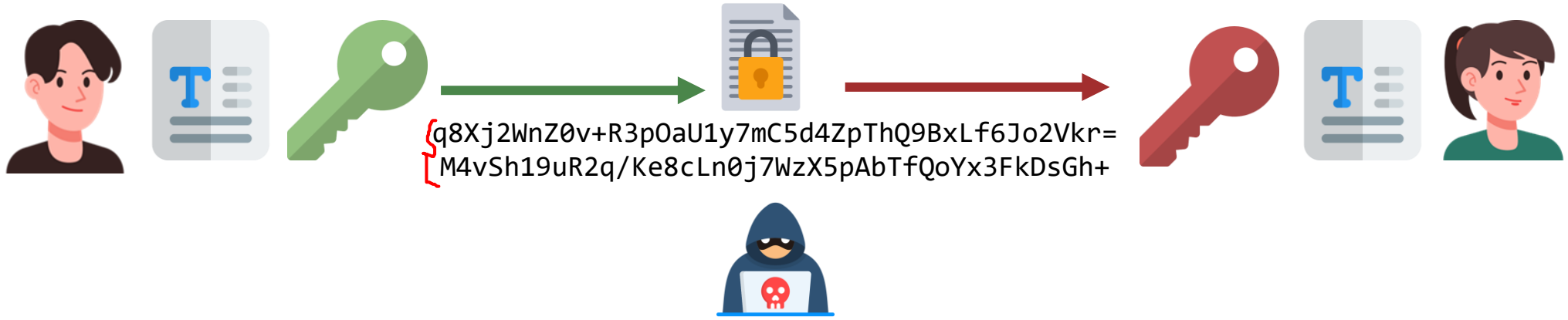


HTTP e HTTPS

Um site que usa apenas o protocolo HTTP envia mensagens em texto limpo.



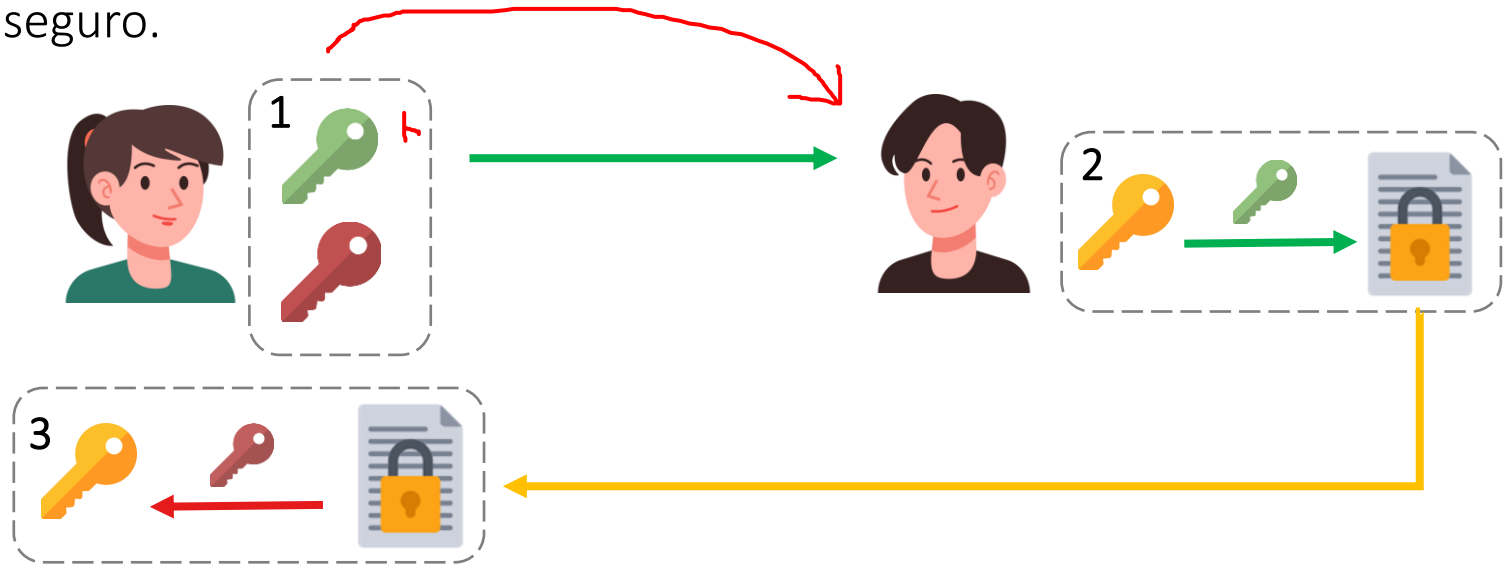
Um site que usa o **protocolo HTTPs** exige que usuários codifiquem mensagens com chave pública, e decifra com chave privada.



Encapsulamento de chave (KEM)

O **encapsulamento de chave** combina criptografia assimétrica (lenta, mas segura na rede) com criptografia simétrica (rápida, mas insegura na rede).

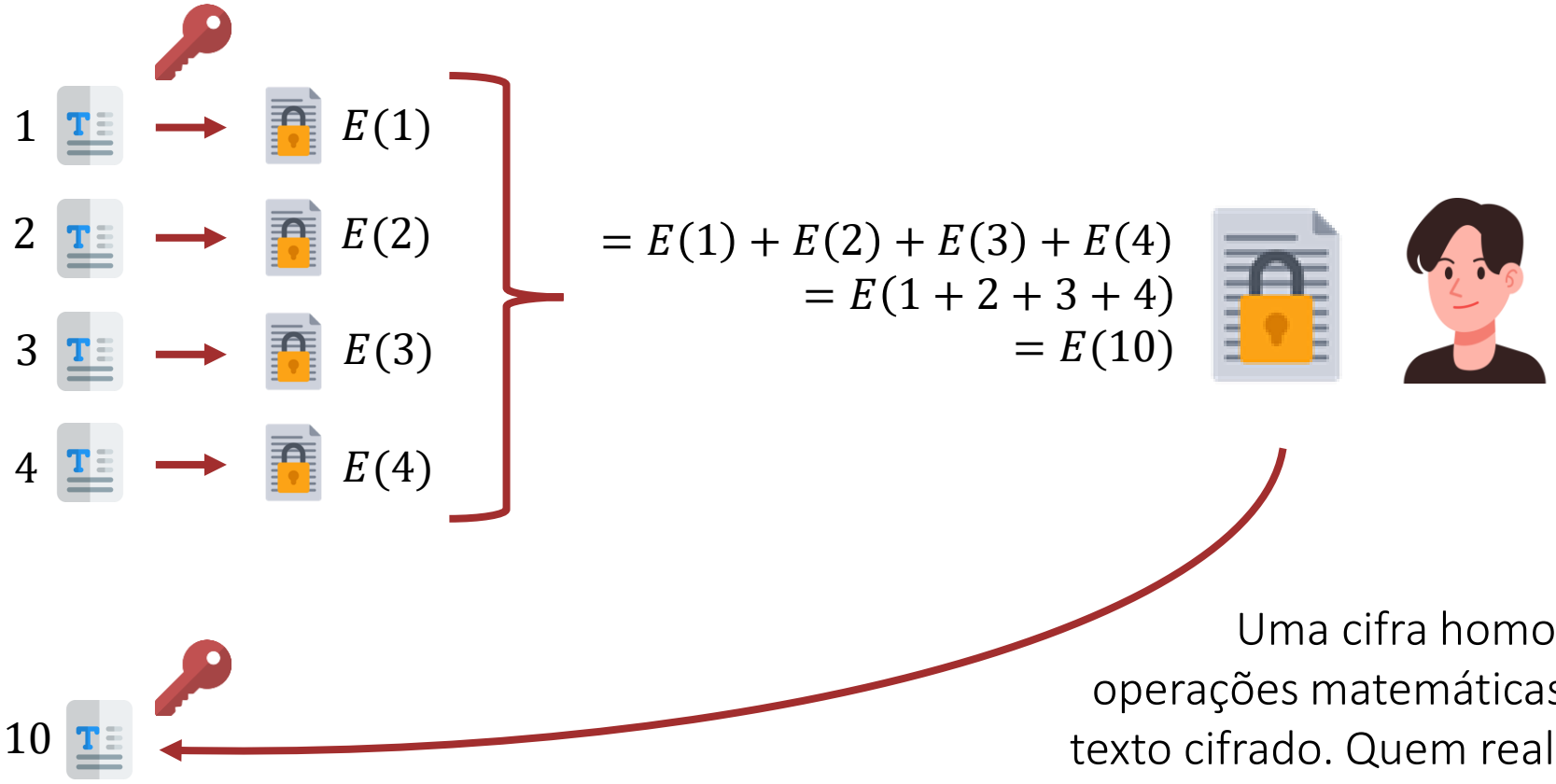
Primeiro, Alice gera um par de chaves assimétricas (1) e envia a chave pública para Bob, que irá cifrar (encapsular) uma chave simétrica (2). Alice irá decifrar a mensagem, e irá receber a chave simétrica (3) que criará um canal simétrico e seguro.



Alice e Bob utilizam criptografia simétrica sem que a chave tenha sido transmitida em texto limpo.

Homomorfismo

O que fazer quando não se confia em quem irá processar dados secretos?



Uma cifra homomórfica permite que operações matemáticas sejam realizadas no texto cifrado. Quem realiza as operações não precisa conhecer a chave e não irá conhecer a entrada ou a saída.

Sistema de Paillier

O sistema de Paillier implementa uma função homomórfica que preserva adição entre valores cifrados e multiplicação por constantes não-cifradas. Abaixo, uma simplificação.

Pré-inicialização

1. Gere dois grandes números primos p e q.
2. Compute $n = p \times q$.
3. Compute $\lambda = \text{mmc}(p-1, q-1)$.
4. Compute $g = n + 1$.

Chaves

1. Chave privada será (n, λ)
2. Chave pública será (n, g)

Codificação

1. A mensagem é um número m.
2. Encripte a mensagem com $c - g^m \bmod n^2$

Decodificação

1. Compute $L(u) = \frac{u-1}{n}$
2. Decripte a mensagem com $m = \frac{L(c^{\lambda} \bmod n^2)}{L(g^{\lambda} \bmod n^2)} \bmod n$.

Soma

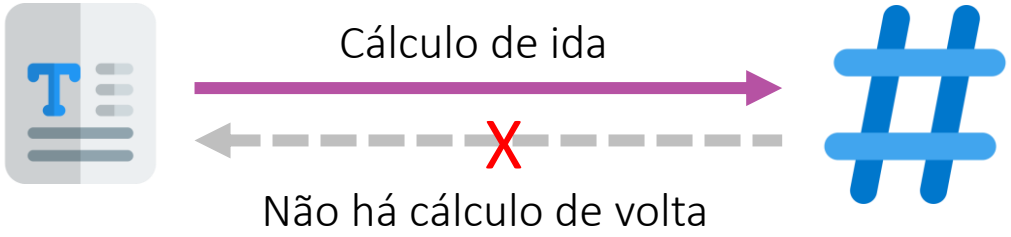
$$m_1 + m_2 = c_1 + c_2$$

Multiplicação por constante

$$k \in \mathbb{R}$$
$$km_1 = kc_1$$

Funções *hash*

Uma **função *hash*** transforma um valor de tamanho arbitrário em um valor de tamanho fixo.



Dois valores podem mapear para uma mesma *hash*, e isso é chamado de colisão.

Ao lado, há uma função *hash* arbitrária extremamente simples.

Cálculo da função

1. Compute o valor ASCII de cada letra.
2. Compute a soma s de todos os valores.
3. Compute $h = s \bmod 23$.

Exemplo

$$h(sapo) = (115 + 97 + 112 + 111) \bmod 23 = 435 \bmod 23 = 21$$

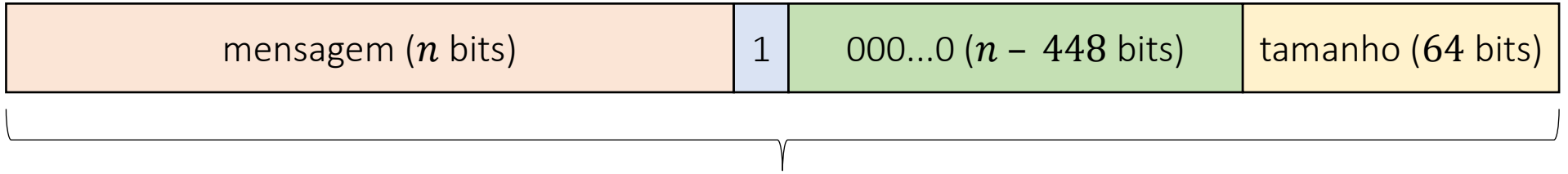
$$h(rato) = (114 + 97 + 116 + 111) \bmod 23 = 438 \bmod 23 = 1$$

$$h(ator) = (97 + 116 + 111 + 114) \bmod 23 = 438 \bmod 23 = 1$$

Não existe forma de reverter uma *hash*, já que múltiplas entradas mapeiam para uma mesma saída.

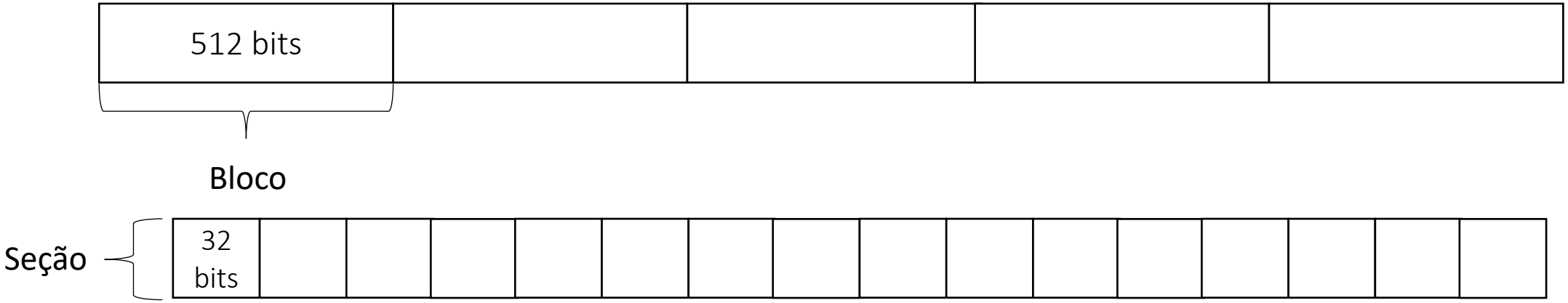
MD5

MD5 é uma função hash relativamente simples e popular, mas já ultrapassada. Primeiro, a mensagem é ajustada para possuir um tamanho que seja múltiplo de 512.



$$n \equiv 0 \bmod 512$$

A mensagem é dividida em b blocos de 512 bits cada. Cada bloco é subdivido em 16 seções de 32 bits cada, chamadas $m_{0\ldots15}$



MD5

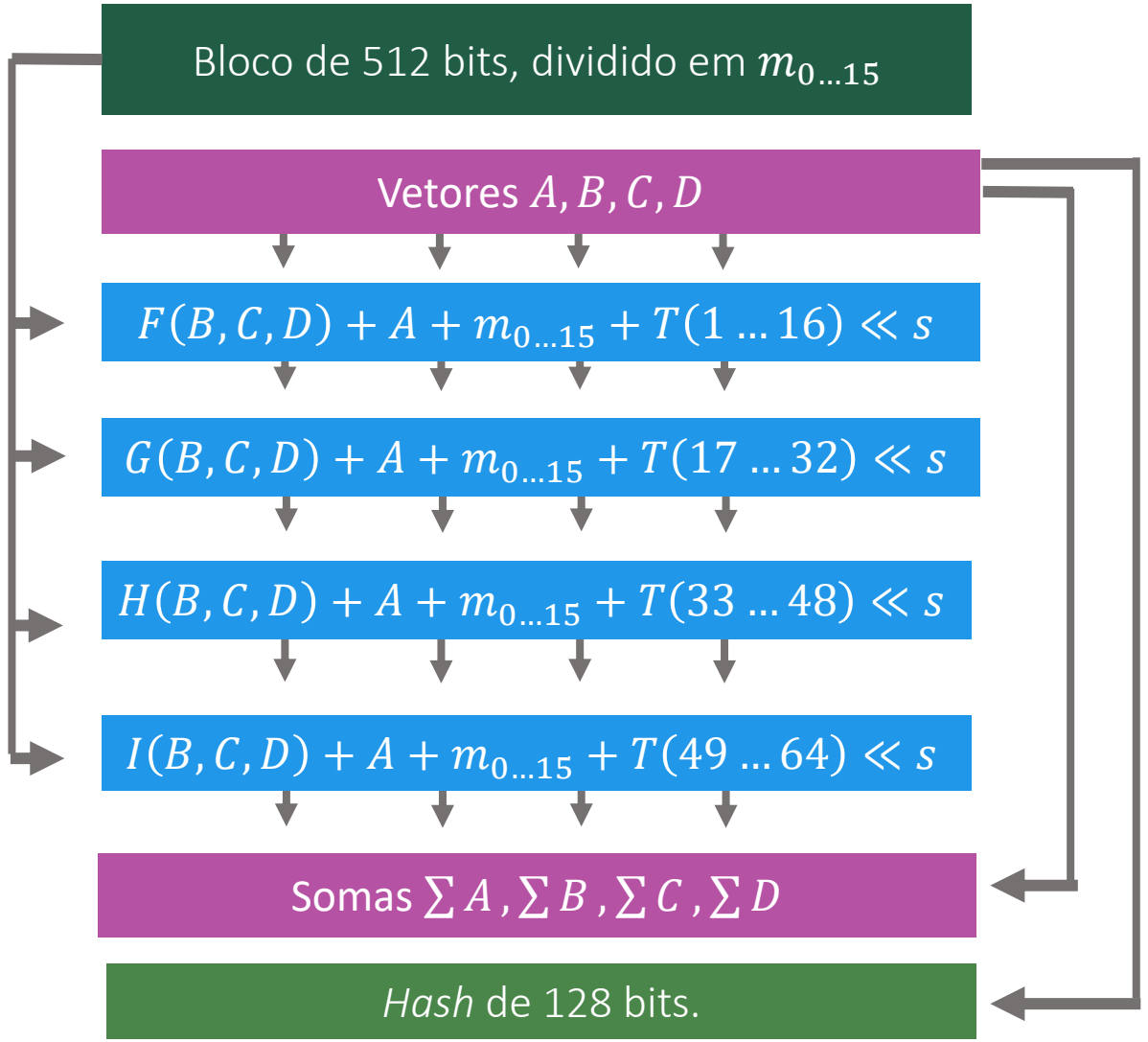
São realizadas 64 operações, sendo 16 de cada tipo.

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge \mathrm{Z})$$
$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg \mathrm{Z})$$
$$H(X,Y,Z) = X \oplus \mathrm{Y} \oplus \mathrm{Z}$$
$$I(X,Y,Z) = Y \oplus (X \vee \neg \mathrm{Z})$$

Para cada uma, somamos um vetor de constantes da forma:

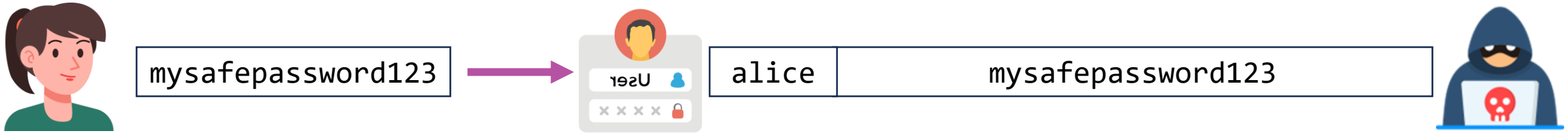
$$T = |\mathrm{sen}(i+1)| \times 2^{32}$$

E usamos um deslocamento à esquerda definido por s.

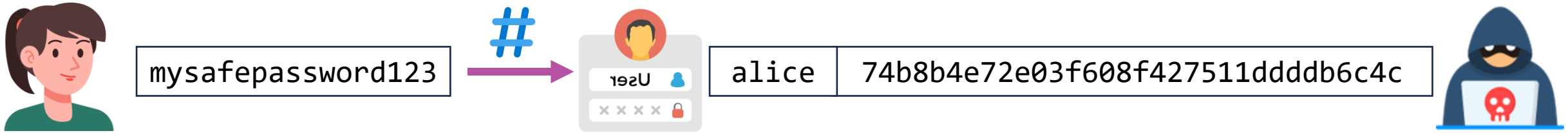

Bloco de 512 bits, dividido em $m_{0\ldots15}$
Vetores A, B, C, D
$F(B,C,D) + A + m_{0\ldots15} + T(1\ldots16) \ll s$
$G(B,C,D) + A + m_{0\ldots15} + T(17\ldots32) \ll s$
$H(B,C,D) + A + m_{0\ldots15} + T(33\ldots48) \ll s$
$I(B,C,D) + A + m_{0\ldots15} + T(49\ldots64) \ll s$
Somas $\sum A, \sum B, \sum C, \sum D$
Hash de 128 bits.

Armazenamento de senhas

Sites são frequentemente comprometidos e, por isso, é uma péssima ideia armazenar senhas em texto limpo. Afinal, um invasor terá acesso imediato à senha.

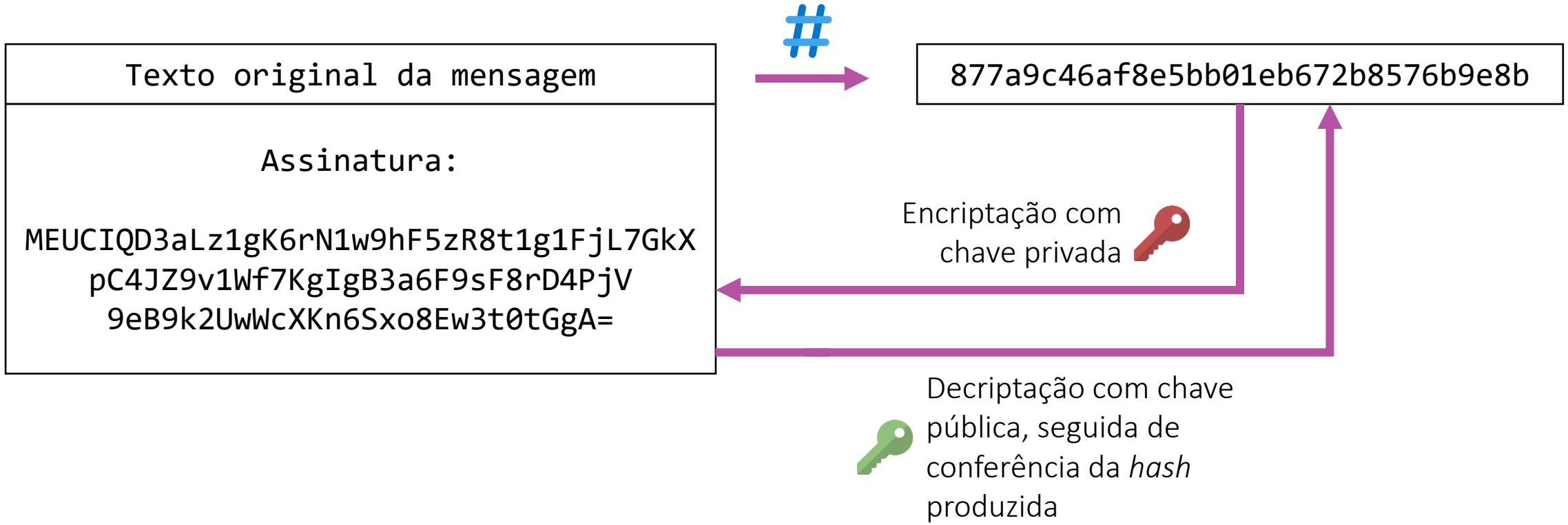


Hashes, por outro lado, trazem uma camada a mais de segurança caso haja um vazamento, pois uma hash é irreversível, e um atacante precisará tentar um ataque de força bruta (vulnerabilidade da senha ou do algoritmo), ou explorar uma colisão (vulnerabilidade do algoritmo).



Assinatura digital

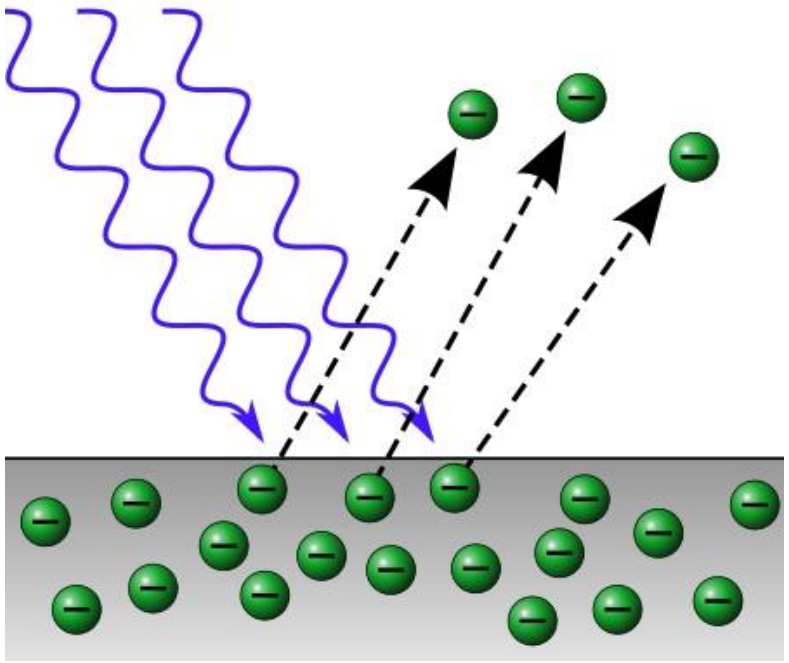
Combinando-se hash e criptografia assimétrica, pode-se criar um mecanismo para garantir que uma mensagem de fato foi enviada por um remetente.



Efeito fotoelétrico

Uma série de avanços no último século provavelmente dará origem a computadores quânticos neste século, e isto levará a mudanças importantes na criptografia.

Einstein demonstrou um efeito constatado décadas antes: **a luz se comporta como partícula**.



Não importa por quanto tempo uma luz de baixa frequência seja incidida sobre um metal, um elétron jamais saltará.

Porém, quando uma luz de alta frequência é incidida sobre um metal, um elétron salta imediatamente.

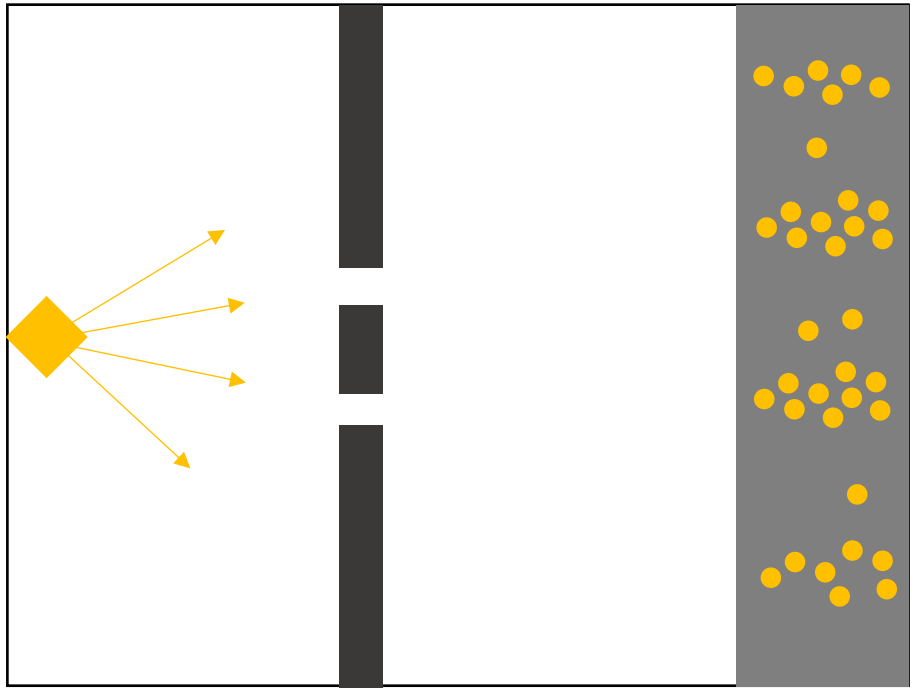
Einstein concluiu que a luz transmite energia ao metal em pacotes discretos.

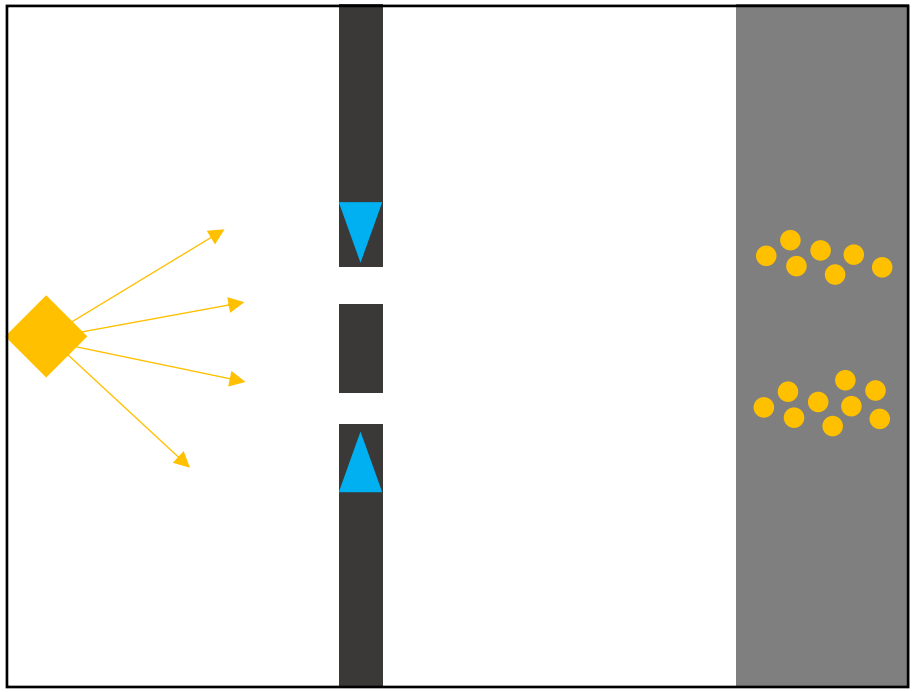
Luz de baixa frequência implica em uma colisão com energia insuficiente, ao passo que luz de alta frequência transmite a energia necessária.

Dualidade onda-partícula

Descobriu-se que não só a luz, mas que **toda matéria se comporta tanto como onda quanto como partícula**, e que essa transição ocorre quando interagimos com a matéria.

Medição sem interação





Medição com interação

Quando uma partícula é emitida e interage somente com um anteparo, ela forma um padrão de interferência decorrente da dupla fenda, tal qual uma onda faria. Ao interagir com um medidor na dupla fenda, ela forma dois aglomerados, como uma partícula pontual faria.

Função de onda

A **função de onda é uma função matemática que expressa probabilidades para o valor de uma grandeza**.
Considere o lançamento de uma moeda honesta.

Antes da moeda ser observada, sua função de onda é:

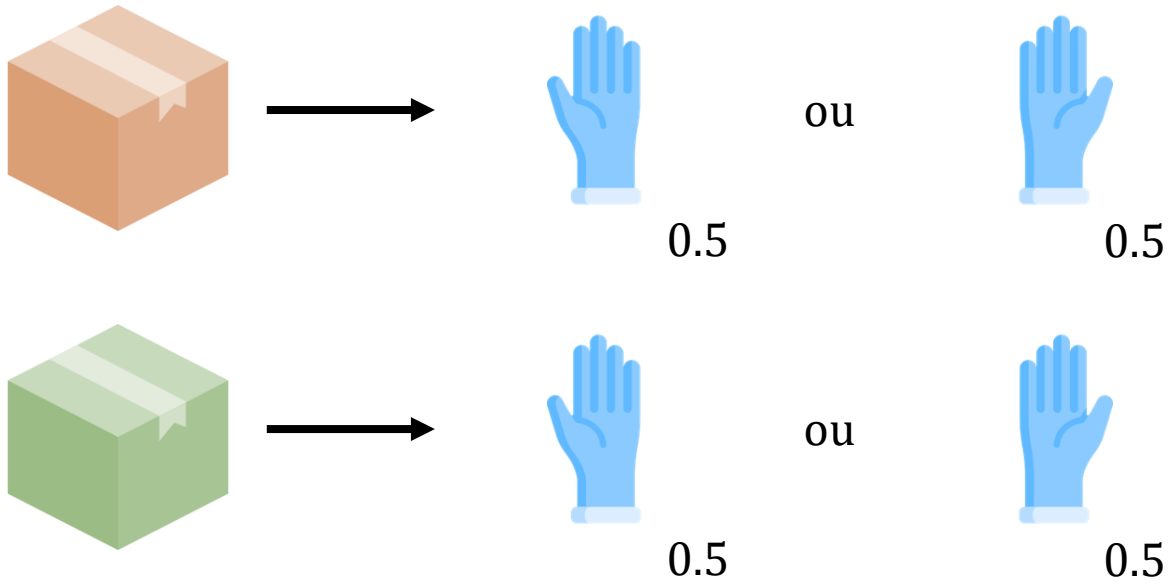
$$|\psi\rangle = \alpha|\mathrm{K}\rangle + \beta|\mathrm{C}\rangle$$

Em que $|K\rangle$ e $|C\rangle$ denotam os estados cara e coroa, e α e β, quando elevados ao quadrado, produzem a probabilidade de se ter cara ou coroa. Como $|\alpha|^2 = |\beta|^2 = \frac{1}{2}$, temos:

$$|\psi\rangle = \frac{1}{\sqrt{2}}|\mathrm{K}\rangle + \frac{1}{\sqrt{2}}|\mathrm{C}\rangle$$

Entrelaçamento quântico

Duas partículas estão **entrelaçadas quando compartilham a mesma função de onda**. Por exemplo, imagine que temos duas caixas, cada uma com uma luva. A função de onda de cada caixa irá representar probabilidades iguais dela conter uma mão esquerda ou direita.



Porém, se as luvas são do mesmo par, elas são entrelaçadas e, logo, abrir uma caixa colapsa a função de onda não só dessa caixa, mas também da outra de forma instantânea.

Bit quântico (*qubit*)

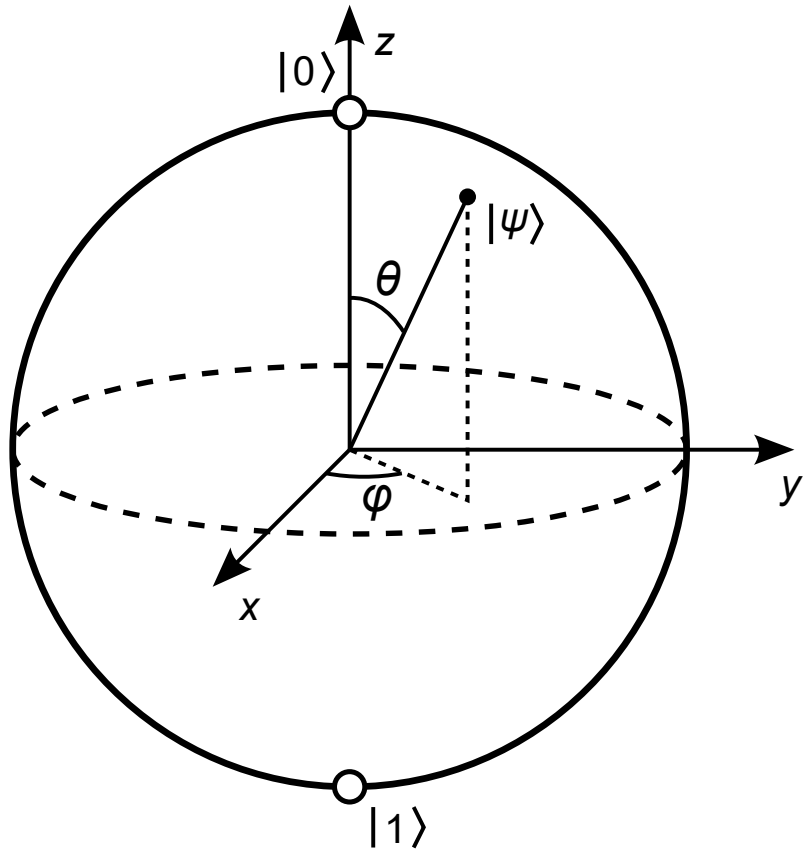
Um bit clássico assume dois valores, 0 ou 1, e seu estado desconhecido reflete ignorância sobre o sistema e não é confiável para computação.

Um *qubit* assume dois valores, 0 ou 1, mas existe em estado de superposição. Sua função de onda é:

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

Para fins práticos, um *qubit* é descrito na forma polar e representado na esfera de Bloch.

$$|\psi\rangle = \cos\frac{\theta}{\sqrt{2}}|0\rangle + e^{i\varphi}\,\mathrm{sen}\,\frac{\theta}{\sqrt{2}}|1\rangle$$



Computação quântica

Com $n = 3$ bits, somos capazes de produzir $2^3 = 8$ estados, mas um por vez. Com $n = 3$ *qubits*, os 8 estados existem simultaneamente em superposição quântica.

$$|\psi\rangle = \alpha_{000}|000\rangle + \alpha_{001}|001\rangle + \alpha_{010}|010\rangle + \cdots + \alpha_{111}|111\rangle$$

Podemos construir portas lógicas e algoritmos que exploram tanto *qubits* em sobreposição quanto entrelaçados que irão:

- Havendo superposição e entrelaçamento, explorar possibilidades de paralelismo inexistentes na computação clássica;

- Sendo funções de onda, interagir de forma destrutiva em saídas indesejadas e de forma construtiva em saídas desejadas.

A **criptografia pós-quântica** propõe algoritmos que sejam seguros mesmo contra computadores quânticos, já que certos algoritmos quânticos, como o algoritmo de Shor, permite quebrar criptografia assimétrica com eficiência muito maior que algoritmos clássicos.

Kyber

AES é resistente a computadores quânticos, mas RSA não. **Kyber** gera um par de chaves para encapsulamento – permitindo que a chave simétrica AES seja transmitida com segurança.

Para isso, primeiro definimos um inteiro positivo q e uma matriz 2×2 de polinômios de grau 256.

Em seguida, geramos tanto um vetor de segredo s contendo 2 polinômios de grau 256 como um vetor de erros e da mesma estrutura.

Então, computamos:

$$b = A \cdot s + e \bmod q$$

A chave pública será (b, A), ao passo que a chave privada será s.

Quando um vetor de erros e é adicionado, recuperar s é caro até para computadores quânticos.

Fim.

Muito obrigado pelo convite e pela presença!